DEPARTMENT OF APPLIED ECONOMICS

RESEARCH REPORT

PROJECT SCHEDULING - THEORY AND PRACTICE

WILLY HERROELEN

OR 0417

Project scheduling - Theory and practice*

Willy Herroelen
Department of Applied Economics, K.U.Leuven
Naamsestraat 69, B-3000 Leuven (Belgium)
e-mail: willy.herroelen@econ.kuleuven.ac.be

7th May 2004

Abstract

The project scheduling problem involves the scheduling of project activities subject to precedence and/or resource constraints. Of obvious practical importance, it has been the subject of intensive research since the late fifties. A wide variety of commercialized project management software packages have been put to practical use. Despite all these efforts, numerous reports reveal that many projects escalate in time and budget and that many project scheduling procedures have not yet found their way to practical use. The objective of this paper is to confront project scheduling theory with project scheduling practice. We provide a generic hierarchical project planning and control framework that serves to position the various project planning procedures and discuss important research opportunities, the exploration of which may help to close the theory-practice gap.

Keywords: project scheduling, commercial software packages, research agenda.

1 Introduction

Scheduling and sequencing is concerned with the optimal allocation of scarce resources to activities over time. The project scheduling problem involves the scheduling of project activities subject to precedence and/or resource constraints. Of obvious practical importance, it has been the subject of extensive research since the late fifties, with an impressive amount of literature as the result. For an extensive review of the literature, we refer to Demeulemeester and Herroelen (2002). Over the years, a wide variety of commercialized project management software packages have been released and put to use in practical project management settings. Despite all these efforts, many publications have

*Background text for the plenary address delivered at the Ninth Project Management and Scheduling Workshop (PMS2004), Nancy, April 26-28, 2004.

appeared in recent years (e.g. Schonberger (1981), The Standish Group (1994), Winch (1996), Yourdon (2003)) documenting projects that went wildly over budget or dragged on long past their originally scheduled completion date. It appears that many scheduling procedures published in the open literature have not yet found their way to the commercial planning software and are seldom or not used by practicing project schedulers. The objective of this paper is to confront the current state of project scheduling theory with current project scheduling practice in general and the use of commercial project scheduling software in particular. We provide a generic hierarchical project planning and control framework that serves to position the various project planning procedures and discuss important research opportunities, the exploration of which may help to close the theory-practice gap.

The organization of this paper is as follows. The next section identifies the need for effective and efficient project scheduling procedures. We discuss the results of various field studies and surveys held among project planning practitioners and put them in perspective with the use and performance of the currently available commercial project planning software and the type of project planning problems that have been the major subject of research over the past several years. Section 3 presents a hierarchical framework for project planning and control that allows at each of its levels to position suitable project planning tools. In Section 4 we focus on the classical deterministic resource-constrained project scheduling problem. In reviewing the state of the art of both exact and suboptimal solution procedures, we reveal some promising future research tracks. We also address the issue of the validation of exact and heuristic solution methods. We identify the risk of overtuning and make some suggestions for remedial action. Section 5 concentrates on the management of project uncertainty. We confront the methodology of stochastic resource-constrained project scheduling with the proactive/reactive scheduling approach, revisit the merits and pitfalls of the critical chain methodology and conclude with the treatment of unknown unknowns. We conclude the paper with a summary and conclusions.

2 The need for effective and efficient project scheduling procedures

2.1 Project management constructs and project escalation

Previous research has documented that projects are frequently prone to escalation (see e.g. Feldman (1985), Flowers (1996), Glass (1988, 1999), Hall (1982), Kharbanda (1983), Morris and Hough (1988)). The often cited reasons why projects escalate are numerous: inadequately trained and/or inexperienced project managers; failure to set and manage expectations; poor leadership at any and all levels; failure to adequately identify, document and track requirements; poor plans and planning processes; poor effort estimation; cultural and ethical misalignment; misalignment between the project team and the business or other organizations it serves; inadequate or misused methods; inadequate communication, including project tracking and reporting (Gantthead (2003); see also Johnson et al. (2001), Zwikael and Globerson (2004)).

A number of authors have made efforts to construct a framework for *classifying critical success/failure factors.* The field study performed by Pinto and Prescott (1990) among 408 managers involved in projects, indicates that critical success factors often fall into two distinctive groups: those related to initial planning (defining the project mission, developing project/schedule plans, client consultation and client acceptance) and those concerned with tactical operationalization (top management support, personnel, technical tasks, monitoring and feedback, communication, and trouble shooting). They found that if traditional measures of project success are employed (budget and schedule adherence), planning issues are most important early in the project implementation process while tactical factors become increasingly important during the later stages of the project lifecycle (see also Pinto and Slevin (1988)). Pinto and Mantel (1990) identify the lack of adequate contingency planning and response development as the single most important factor in predicting project failure (see also Pinto (2002)).

Belassi and Tukel (1996) grouped the factors into four areas: factors related to the project (size and value, uniqueness of the project activities, project density, life cycle and urgency), factors related to the project manager and the team members (ability to delegate authority, ability to trade-off and coordinate, perception of responsibilities, competence, commitment, technical background, communication skills, trouble shooting), factors related to the organization (top management support, project organizational structure, functional managers' support, project champion), and factors related to the external environment (political, economical, social and technological environment; nature; client; competitors and sub-contractors). The results of their empirical study show a noticeable shift in ranking from organizational factors towards factors related to project managers and team members and towards the factors related to projects.

The field research conducted by Milis and Mercken (2002) among Belgian banks and insurance companies allowed them to identify four categories of factors: factors that influence goal congruency (project selection, scope definition, definition of success criteria), factors that are project team related (goal setting, communication and conflict control, technical and social skills, etc.), factors that relate to the acceptance of the project and its results (top management support, training, competent project manager, etc.), and factors related to the implementation process and planning (sufficient resources, change management and contingency planning, built in resource buffers).

Keil et al. (2003) have gathered data on information system projects (579 surveys received from IS audit and control professionals in the U.S.) and applied logistic regression to model the relationship between various project management constructs and project escalation. The key constructs included project planning (definition of project goals and objectives, setting out the deadlines for completion of the deliverables, development of the project schedule, development of success criteria for the project deliverables, development of contingency plans on potential risk factors), project specification (specification of user needs, scope management, definition of accurate specifications), project estimation (developing activity duration and resource requirement estimates, estimating project completion time), and project monitoring and control (capture, analysis and reporting of project performance, taking corrective actions, taking preventive action in anticipation of possible problems). The variables that best distinguished between escalated and nonescalated projects were found to be monitoring and control. Estimation and specification were deemed significant. Planning exhibited a moderately high correlation with monitoring and control. The authors reached the conclusion that managers who are willing to invest in project management tools, techniques and training, particularly in the areas of specification, estimation, monitoring and control could reap substantial benefits in terms of reducing the incidence in runaway projects.

Söderlund (2004) believes that a certain tradition of success-oriented research has dominated and argues that the major part of critical success factor research does not give us deeper knowledge about real life project management. His main argument is that this research does not acknowledge the dynamics and the social embeddedness of project management, and that there are openings for additional perspectives and empirical studies. Bryde (2003) conducted an empirical study of project management practice and of attitudes and opinions of people involved in projects in UK organizations to determine the extent to which project management has evolved into being broader in its concepts, methods and application than "traditional" project management. Overall it is concluded that project management practices confirming a broadening of the application, concepts and methods of project management were, at best, variable and patchy, being present in some organizations and not in others. Applications of project management regarded as most useful were still those traditionally associated with project work in such areas as construction, engineering or the introduction of new systems, rather than those concerned with the management of all business-related change.

2.2 The use of software tools

The conclusions that can be drawn from studies on the usage of project management software are far from consistent. Bounds (1998) concluded from a reader survey among IIE members that 80% of the respondents used some kind of project planning software. When asked to rank the capabilities that were most important in a tool of this type, respondents overwhelmingly chose project tracking first (28 percent ranked it No. 1 and 24 percent ranked it No. 2) and time analysis second (20 percent ranked it No. 1 and an additional 20 percent ranked it No. 2). Third in importance was cost analysis (17 percent ranked this No. 1, 9 percent ranked this No. 2). As for resource analysis, only 12 percent ranked it No. 1 and 14 percent ranked it No. 2. This stands in sharp contrast with the importance adhered to resource-constrained project scheduling in academic writings. The industrial engineers identified ease of use and project tracking as the most critical project management tool features.

Pollack-Johnson and Liberatore (1998) have conducted a survey among 688 randomly chosen members of PMI (Project Management Institute). A total of 240 responses revealed that almost all project management professionals used project management software to some extent. Of those using software, about 95 % use it for planning and 80 % for control (Liberatore and Pollack-Johnson (2003), Liberatore et al. (2001)). The results of the above mentioned surveys stand in strong contrast to recent surveys conducted in Europe. Both the surveys conducted by De Reyck and van de Velde (1999) among companies operating in various industrial sectors in the Netherlands (infrastructure, construction, software engineering, product and process design and maintenance) and Deckers (2001) among similar companies operating in Belgium, reveal that (a) information systems for project planning are mainly used for communication and representation, rather than for optimization, and

(b) that software users have a limited knowledge of the software tool they are using and of project planning tools in general. These results provoke the prudent conclusion that proper use of project management software may well not be considered the most important driving force behind project success.

From the previous mentioned studies, it appears that the two most popular project management software packages are Microsoft Project and Primavera Project Planner. Johnson et al. (2001), referring to the "Extreme CHAOS 2001" report by The Standish Group, state that IT project success rates are up, overruns are down, with a substantially declining rate of failure. They adhere an important role to two types of project management tools: *professional service automation* (*PSA*) and *enterprise project management* (*EPM*). *PSA* tools (e.g. Evolve, Evolve Software, Inc. (www.evolve.com); PlanView, Inc. (www.planview.com)) form a suite of software modules or applications that together can handle multiple projects and contributors (e.g. staff and contractors). *PSA* focuses on optimizing service processes for acquiring, managing, and fulfilling service engagements external to the enterprise. The heart of most *PSA* tools are the resource management modules that are used to staff engagements based on available resource skill, using a common repository of staff skills and historic project information. It has been estimated that by 2005, 40 percent of Global 2000 organizations will employ enterprise program management (*EPM*) to implement enterprise solutions underlying technical architecture and infrastructure (Bigelow (2004)). *EPM* software modules (e.g. Artemis Management Systems (www.artemispm.com); Primavera (www.primavera.com)) are most often used to manage the enterprise organization's internal multiple projects. They should allow organizations to establish business processes and project priorities, mapping them against corporate objectives. Through project portfolio management and analysis, managers should be able to maximize resource utilization, eliminate duplicate projects, collect organizational knowledge and institute best practices. Team members should be able to benefit from better communication, streamlined resource management and improved productivity.

2.3 The project scheduling software performance

Various authors have evaluated the quality of project management software. The studies can be divided into two groups. The first group focused on the general software capabilities (De Wit and Herroelen (1990) and Maroto and Tormos (1994)), while the second group evaluated the quality of the generated resource-constrained project schedules (Johnson (1992), Kolisch (1999), Maroto et al. (1999)).

De Wit and Herroelen (1990) pointed out that the jargon used by the professional project planning software basically neglected the standardized terminology that is internationally accepted among teachers and researchers in the field. This continues to be the case. The problem solving methods are proprietary and mostly not revealed. Most commercial software packages do not embody exact algorithms for resource levelling and resource-constrained scheduling, but rely on the use of simple priority rules for generating a precedence and resource feasible schedule. These rules assign scheduling priority on the basis of activity attributes such as latest start time (LST), latest finish time (LFT), minimum slack (MINSLK), etc. Some packages enable the user to select a priority rule from a (sometimes very extensive) list, while others do not. We are not aware of commercial software packages capable of dealing with other resource analysis problems than the classical resource-constrained scheduling problem and the resource levelling problem.

Maroto et al. (1999) use makespan as the performance measure for single projects and mean project delay (average increase in project completion time with respect to its initial critical path) as the performance measure for multi-projects. Using a testset of 96 resource-constrained project instances with up to 50 activities, they found that CA-Superproject and Time Line are the best performers in terms of makespan, even better than more expensive software, such as Artemis Schedule Publisher and Primavera. The best performing packages had a makespan performance similar to that of LFT. A testset of 32 multi-projects revealed that Primavera Project Planner was the best performer, with a performance similar to the SASP (shortest activity, shortest project) and MAXTWK (maximal total work content) priority rules.

The computational experiment performed by Kolisch (1999) with seven commercial software packages on 160 test instances, reveal that Primavera Project Planner delivers the best resource-constrained project scheduling performance, especially in a multi-project environment. Average performance was variable, with the best package deviating on the average 4.39% and the worst package deviating on the average 9.76% from the optimum makespan. The mean deviation from the optimum makespan is 5.79%, while the standard deviation calculates to 7.51% and the range is from 0 to 51.85%. The scheduling performance of commercial software decreases with increasing number of activities, increasing number of requested resource types, and with decreasing resource capacity. The packages produce better results for projects with many precedence relations.

Until recently, commercial software packages generated deterministic baseline schedules without any protection against uncertainty (although the possibility to perform a *risk analysis* based on Monte Carlo simulation has been incorporated in many software packages, e.g. Risk+ (C/S Solutions (2002)). The *critical chain* methodology (Goldratt (1997)) paved the way for the recent incorporation of protective buffering mechanisms into the planning software. The software that implements the critical chain methodology comes as a software add-in for popular commercial project planning software packages such as Microsoft Project, or is part of the original project planning software package. In the first category we list ProChain (for information we refer to www.prochain.com), cc-Pulse and cc-MPulse (information can be found at www.sphericalangle.com), and Realization (Concerto) (information at www.realization.com). An example of the second category is PS Suite (PS8, Project Communicator, PSI) and PSNext (for information, we refer to www.sciforma.com). These software packages allow for the generation of a baseline schedule that contains time buffers that should serve as a proactive protection mechanism against disturbances that may occur during project execution.

2.4 Type of projects

Most of the project scheduling research efforts over the past several years have focused on the scheduling of the activities of a *single project*. For an extensive overview and discussion of the scheduling procedures we refer to recent books such as Demeulemeester and Herroelen (2002), Klein (2000), Neumann et al. (2003), and Weglarz (1999), and various survey papers such as Özdamar and Ulusoy (1996), Brucker et al. (1999), Herroelen et al. (1997), Herroelen et al. (1998a), and Kolisch and Padman (2001).

The literature on the simultaneous scheduling of *multiple projects* is rather sparse. In a first approach, the projects are artificially bound together into a single project by the addition of two dummy activities representing the start and end of the single 'aggregate' project, possibly with different ready (arrival) times and individual due dates. In such a case, existing exact and suboptimal procedures for single-project scheduling may be used for planning the aggregate project.

In a second approach, the projects are considered to be independent and specific multi-project scheduling techniques - mostly heuristic in nature - are used. Kurtulus and Davis (1982) report on computational experience obtained with six priority rules under the objective of minimizing total project delay. Kurtulus (1985) and Kurtulus and Narula (1985) analyze the performance of several priority rules for resource-constrained multi-project scheduling under equal and unequal project delay penalties. Lova et al. (2000) have developed a multi-criteria heuristic for multi-project scheduling for both time-related and time-unrelated criteria. Lova and Tormos (2002) have developed combined random sampling and backward-forward heuristics for the objectives of mean project delay and multi-project duration increase.

Several authors have studied the problem of *assigning due dates* to the projects in a multi-project environment. Dumond and Mabert (1988) evaluated the relative performance of four project due date heuristics and seven resource allocation heuristics; related research can be found in Dumond (1992). Bock and Patterson (1990) investigate several of the resource assignment and due date setting rules of Dumond and Mabert (1988) to determine the extent to which their results are generalizable to different project data sets under conditions of activity preemption. Lawrence and Morton (1993) study the due date setting problem and performed large-scale testing of various heuristic procedures for scheduling multiple projects with weighted tardiness objective. Several model extensions are discussed in Morton and Pentico (1993).

The fact that most project scheduling research has focused on single project scheduling (to illustrate, only some 10 presentations on multi-project planning have been made at the PMS (Project Management and Scheduling) workshops since the first workshop organized by the EURO Working Group on Project Management and Scheduling in Lisbon in 1988), stands in strong contrast with the conclusions emerging from recent surveys among project management practitioners. Liberatore et al. (2001) and Liberatore and Pollack-Johnson (2003) have deducted from their survey that project management has to face mostly up to four projects each containing more than 150 activities. Icmeli-Tukel and Rom (1998) concluded that a typical project consists of some 100 activities, while management typically has to control more than 5 projects simultaneously and has to deal with 2-5 subcontractors. Maroto et al. (1999) conclude that management mostly operates in a multi-project setting, where most projects are limited in size (up to 50 activities with 1-5 nodes in parallel).

Multi-project environments seem to be quite common in project scheduling practice and offer many future research opportunities. It has been suggested (Payne (1995), Turner (1993)) that up to 90%, by value, of all projects are carried out in the multi-project context, and thus the impact of even a small improvement in their management on the project management field could be enormous. In the next section we discuss a hierarchical planning and control framework, that allows at each of its hierarchical levels for the application of a positioning framework to position suitable project planning procedures.

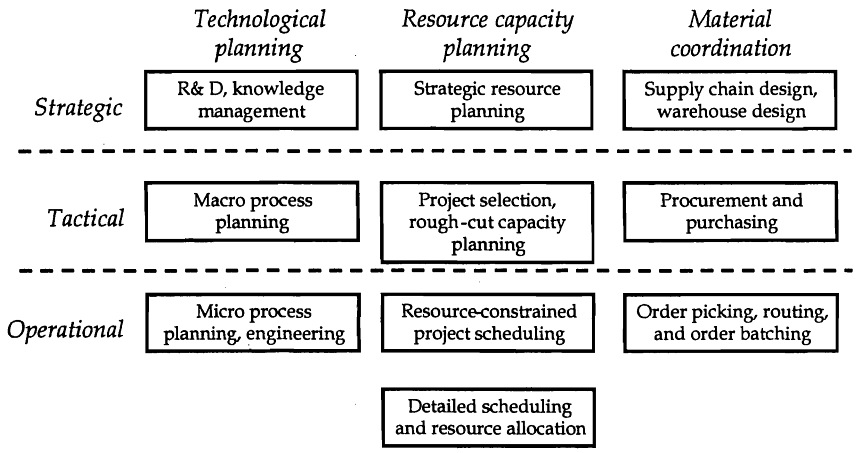

Figure 1: Hierarchical framework

3 A hierarchical framework for project planning and control

Figure 1 shows a hierarchical planning and control framework (Hans et al. (2003)) that distinguishes three hierarchical levels (the strategic level, the tactical level, and the operational level), and three functional planning areas (technological planning, capacity planning, and material coordination). Four resource capacity planning functions are identified: (a) strategic resource planning, (b) rough-cut capacity planning, (c) resource-constrained project scheduling, and (d) detailed scheduling. In this paper we focus on the tactical and operational levels, more in particular on the capacity planning, scheduling and resource allocation issues.

At each level of the hierarchy, we can apply the positioning framework shown in Figure 2. The objective of this framework is to position project planning methods using two key determinants: the degree of variability in the work environment and the degree of dependency of the project (Leus (2003), Herroelen and Leus (2004b)).

The *variability* is an aggregated measure for the uncertainty caused by the lack of information at the tactical stage (detailed information about the required activities usually becomes only gradually available) and the operational uncertainties on the shop floor. It involves a joint impression of the uncertainty and variability associated with the size of the project parameters (time, cost, quality), uncertainty about the basis of the estimates (activity durations, work content), uncertainty about the process (what is to be done, how, by whom and at what cost), uncertainty about the objectives, priorities

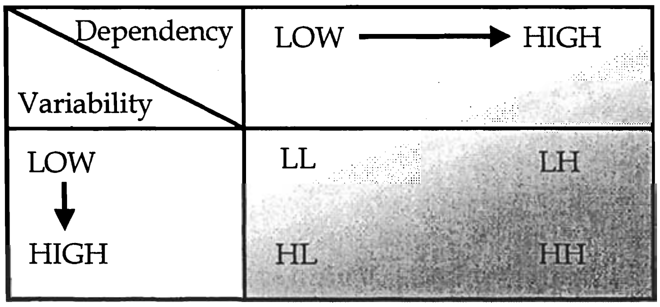

Figure 2: A positioning framework for multi-project organizations

and acceptable trade-offs, and uncertainty about fundamental relationships between the various project partners involved (responsibilities, communication, contractual conditions and effects, mechanisms of coordination and control). The *dependency* measures to what extent a particular project is dependent on external influences. These influences can be actors from outside the company (e.g. subcontractors or material coordination), but also dependencies from inside, for instance shared resources with other projects. We assume that the scale of the dimensions in Figure 2 is continuous. For reasons of simplicity we discuss the four extreme cases of low and high variability and low and high dependency, where it should be understood that all possible intermediate positions in between the four extreme cases are conceivable. We distinguish between:

- *LL (Low variability and low dependency).* Low variability and low dependency is typical for single-project organizations where the resources are completely dedicated to one particular project and activities have a low degree of uncertainty. Hans et al. (2003) refer to on-site preventive maintenance projects for which the activities are often routine and specified in advance so that the degree of uncertainty is relatively low. Moreover, there is little interaction with other projects, so that the degree of dependency is also on the low side. Often projects are executed within a pure project organization. The allocation of resources to projects can already be done at the tactical level. Deterministic single-project scheduling methods can be used to schedule each individual project.

- *LH* (*Low variability and high dependency*). The *LH*-setting is related to the classical make-to-order job shop, for example a small furniture manufacturer that produces wooden furniture on a make-to-order basis. Equipment is general-purpose and may be claimed by different projects so that dependency is on the high side. Projects usually have to be executed within a matrix organization. The manufacturing process is relatively basic. Product complexity and process complexity are low resulting in low variability.

- *HL* (*High variability and low dependency*). Large construction projects are typical for a high variability, low dependency environment. Variability is high due to large environmental uncertainties such as bad weather conditions, uncertain or changing project specifications. The dependency is usually low because the projects are executed within a pure project organization and deployed resources are often dedicated.

- *HH* (*High variability and high dependency*). A high degree of uncertainty in combination with a high degree of dependency can usually be found in engineer-to-order environments where every new product requires a long and intensive engineering process and where changes in customer specification are frequent. In combination with the complexity of the product, an extremely high degree of variability is the result.

3.1 Rough-cut capacity planning

The tactical planning stage is characterized by a sufficient degree of capacity flexibility (e.g. overtime work, subcontracting). Tactical planning therefore requires methods that use more aggregate data, and that exploit this capacity flexibility. Deterministic rough-cut capacity planning procedures have been proposed by De Boer (1998), Hans (2001) and Gademann and Schutten (2001). These procedures use an objective function that minimizes the per period cost of using non-regular capacity (overtime, hiring additional staff and subcontracting). The authors implicitly claim that for project environments in the *LL*- and *LH*-categories it is sufficient to choose a proper data aggregation level to cope with possible disturbances. For *HL*- and *HH*-project environments, however, we believe that the planning procedures should be able to deal with the uncertainties that typically occur at the tactical planning stage. Elmaghraby (2002) and Tereso et al. (2004), for example, claim that the major uncertainty resides in the work content of the project activities. Other types of uncertainty that may occur at the tactical level may relate to the occurrence of activities, resource availability, release and due dates. The literature on project selection and rough-cut capacity planning under uncertainty is rather scarce. Kavadias and Loch (2004) develop a dynamic model of resource allocation, taking into account multiple interacting factors, such as uncertain market payoffs that change over time, increasing or decreasing returns from the investment, carry-over of the investment benefit over multiple periods, and interactions across segments. The authors also derive an optimal admission policy for projects of varying potential reward that arrive at unpredictable points of time. Wullink et al. (2003) propose a proactive rough-cut capacity planning approach that uses a scenario-based MILP-model to minimize the expected costs of using non-regular capacity.

3.2 Information exchange

Project organizations with high dependency (*LH* and *HH*) generally adopt a matrix-organizational structure. Hans et al. (2003) suggest that during the early stages in the project lifecycle, the rough-cut capacity planning procedures provide due dates, milestones and required capacity levels. Together with additional information that becomes available during later stages, these data are passed on as input to the operational level, where multi-project resource-constrained project planning procedures may be used. Projects in the *LL*- and *HL*-environments are usually performed by dedicated or pure project organizations. The assignment of resources to the various projects can already be made at the tactical level, so that these resource allocation decisions can also be passed on to the operational level, where multiple separate single-project plans may be developed.

3.3 Resource-constrained scheduling at the operational level

When dependency and variability are on the low side (*LL*-environment), deterministic single-project scheduling methods can be used to schedule each individual project with dedicated resources. For *HL*-environments, project management may rely on dispatching rules or proactive/reactive scheduling procedures. *LH*-environments call for scheduling procedures for generating robust (stable) schedules to prevent propagation of the small disruptions throughout the overall plan. In *HH*-environments, characterized by high variability and high dependency, a process viewpoint may be taken. Resources are workstations that are visited by work packages that are passed on to successor resources upon completion. A rough ballmark plan should allow to come up with intermediate milestones, used for setting priorities for the resources in selecting the next work package to be processed.

Intermediate cases with moderate dependency may allow for the identification of *drum* activities that induce the uncertainty (Leus (2003)). These activities can be planned first for efficiency and stability, while the remaining activities are scheduled from the start or are dispatched in function of the progress on the drum.

4 Deterministic resource-constrained project scheduling

In the last few decades, several effective algorithms for solving the well-known *single-project deterministic resource-constrained project scheduling problem* (RCPSP) have been proposed (problem $m,1|cpm|C_{max}$ using the classification scheme of Herroelen et al. (1998b)). The problem involves the determination of a precedence and resource-feasible baseline schedule that minimizes the project duration. Numerous exact and heuristic solution procedures have been developed. The problem is that, apart from simple priority rule based heuristics, none of these algorithms found their way to commercial project planning software.

4.1 Exact solution procedures

4.1.1 Branch-and-bound

The most noteworthy exact solution procedures for the RCPSP rely on *explicit branch-and-bound* (Brucker et al. (1998), Demeulemeester and Herroelen (1992, 1997), Mingozzi et al. (1998) and Sprecher (2000)), i.e. the partial enumeration of schedules to create a search tree rather than the use of fractional solutions obtained from the LP relaxation. The efficiency of the explicit branch-and-bound methods heavily depends on the procedure used for constructing/searching the *search tree* and fathoming branches as early as possible. For an extensive discussion, we refer to Demeulemeester and Herroelen (2002). A promising research area is the further exploration of new procedures for *decomposing* the potential search tree into disjoint subtrees that are searched independently and the development of new *search diversification* strategies (see e.g. the *scattered branch-and-bound* procedure developed by Klein (2000)).

There has been a fair amount of work devoted to obtaining good *lower bounds* and dominance rules. We can roughly divide the lower bounds encountered in the literature into three categories: *specific lower bounds* derived from the special structure of the RCPSP (for an excellent review see Klein and Scholl (1999)), *linear programming relaxation based lower bounds* (for example Carlier and Néron (2000), Christofides et al. (1987), Sankaran et al. (1999), Möhring et al. (2003), Mingozzi et al. (1998), Brucker et al. (1998), Brucker and Knust (2000)), and *constraint programming bounds* (Baptiste and Le Pape (2000), Baptiste et al. (1999), Dorndorf et al. (2000)). Among the LP based lower bounds, it is worthy of note that one of the tightest bounds (Brucker and Knust (2000)) is of the destructive type (destructive methods show that there is no solution of total duration smaller than or equal to a given upper bound T, so that $T+1$ is a lower bound). Demassey et al. (2003) propose a new lower bound based on a cooperation between linear programming and constraint propagation. They propose a new shaving technique (setting the starting time of an activity to the left bound of its time window and apply local constraint propagation rules to prove infeasibility of this starting time in which case the time window is reduced and the shaving process is reiterated). Results obtained on the KSD-instances with 30, 60, 90 and 120 activities (Kolisch et al. (1995)) show that generating cutting planes based on deductions performed by constraint propagation can be an alternative to the classical specific lower bounding procedures. The bound is rather time consuming and significantly weaker on average than the Brucker & Knust bound. Baptiste and Demassey (2004) improve the Brucker & Knust bound by using (a) a preprocessing step based on intensive constraint propagation of redundant machine constraints to tighten the initial formulation of the linear programs, and (b) the addition of energetic, non-preemptive and precedence cuts to the MIP problem solved by CPLEX. Other recent research efforts exploring linear programming relaxations include Damay et al. (2004) and Demassey et al. (2004). The obtained results still leave room for additional research and improvement.

4.1.2 Branch-and-cut

Zhu et al. (2003) are the first to apply branch-and-cut to the (multi-mode) resource-constrained project scheduling problem (problem $m,1T|cpm,disc,mu|C_{max}$). Their procedure uses the LP relaxation of the integer linear programming model to obtain a lower bound at each node of the search tree. If a node has a fractional solution and cannot be fathomed, they derive cuts that are violated by the fractional solution but are satisfied by all feasible integer solutions. If no cut can be found, branching is performed to create new nodes in the search tree. The authors use the cut generating features built into the MIP solver that comes with CPLEX (ILOG 2002). In addition they derive problem-specific cuts to tighten the LP bounds. In addition to variable reduction and bound tightening procedures, they use a high level neighborhood search strategy referred to as *local branching* (Fischetti and Lodi (2003)) to find good initial solutions in the early stages of the computations. To the best of our knowledge, computational results obtained with branch-and-cut procedures on the classical RCPSP have not been reported. The development of new approaches for exploring neighbourhoods in mixed integer programming (MIP) constitutes a viable area of research. Danna et al. (2004) have recently proposed two new approaches (Relaxation Induced Neighborhood Search (RINS) and guided dives) that outperform local branching on very difficult MIP problems such as the job shop.

4.2 Heuristic procedures

The RCPSP, being a generalization of the job shop scheduling problem, is strongly NP-hard, and the computation times for exact algorithms can be excessive even for moderately sized instances. This phenomenon has motivated numerous researchers to design heuristic procedures. For a classification and performance evaluation of heuristics focusing on *X-pass methods* (single pass methods, multi-pass methods, sampling procedures) and *metaheuristics* (simulated annealing, genetic algorithms and tabu search), we refer to Kolisch and Hartmann (1999) and Hartmann and Kolisch (2000). The computational results indicate that the best metaheuristics outperform the best sampling approaches.

The main focus of *metaheuristic research* was on the application of *single* metaheuristics. Recent research, however, demonstrates that concentration on a sole metaheuristic is rather restrictive. A skilled combination of concepts of different metaheuristics can provide a more efficient behavior and a higher flexibility. Valls et al. (2003), for example, have developed a *hybrid* genetic algorithm that seems to outperform all state-of-the-art algorithms - at least for the 120 activity KSD instances. Their procedure uses a (peak) crossover operator that is not pure random nor context free, but combines useful problem-specific information extracted from the parents with the purpose of generating high-quality children. The authors use 'double justification' as a simple, fast, and powerful mechanism to improve schedules and rely on a two-phase strategy by which the second phase restarts the evolution from a neighbor's population of the best schedule found in the first phase.

Clearly, the derivation of *hybrid metaheuristics* that incorporate more classical AI and OR techniques such as shrinking the search space by using domain filtering and variable fixing and incorporating tree search procedures, constitutes a viable area of future research. The same can be said for the ingenious combination of scatter search techniques, generic population-based evolutionary search and electromagnetism-based heuristics originally introduced for the optimization of unconstrained continuous functions (Debels et al. (2004)). The exploration of hybrid metaheuristics may allow for breaking through the 31% average percentage deviation from the critical path lower bound barrier, so characteristic for the performance of state-of-the-art heuristics. The results obtained by Fleszar and Hindi (2004) using their variable neighbourhood search procedure confirm that careful control of the trade-off between solution quality and computational requirements remains a crucial issue.

4.3 Generating test instances

Project characteristics have long been overlooked in the literature as being critical success factors whereas they constitute one of the essential dimensions of project performance (Belassi and Tukel (1996)). It is common practice to test solution procedures on a set of test instances generated by problem generators. Ideally, the generators should generate problem ensembles that span the full range of problem complexity (Elmaghraby and Herroelen (1980)) and that can be tuned to fit the unique characteristics of real-world scheduling problems.

The characteristic that the test instances should span the full range of problem complexity is crucial. Researchers in the field of artificial intelligence (Cheeseman et al. (1991), Hayes (1997), Huberman and Hogg (1987)) discovered that many NP-complete problems exhibit so-called *phase transitions*, resulting in a sudden dramatic change in computational complexity. Often, problem instances change from being easy to being hard to solve to again being easy to solve when certain of their characteristics are modified. Most often the transitions are sharp, but sometimes they are rather continuous in the order parameters that are characteristic of the system as a whole. Hard to solve instances are often clustered around a small range of the order parameter values, which implies that most instances (when looking at the entire range of the order parameters) are easy to solve.

Herroelen and De Reyck (1999) have studied the existence of phase transitions in various resource-constrained project scheduling problems and have called the attention to the importance of measures with sufficient discriminatory power to allow for the observation of these dramatic changes in problem difficulty. The popular problem generators (*ProGen* (Kolisch et al. (1995)), *ProGen/max* (Schwindt (1996)), *RanGen* (Demeulemeester et al. (2003)) rely on 'complexity measures' to capture information about the size of the network, the topological structure (morphology) of the project network and the availability of the different resource types in relation to the resource requirements. *GenRes* (Coelho (2004)) generates problem instances that satisfy preset values of resource-based parameters and for which the optimal solutions are known.

4.3.1 Network-based parameters

An often used metric for the network structure is the *coefficient of network complexity* (*CNC*) defined by Pascoe (1966) for activity-on-the-arc networks as arcs over nodes, and subsequently adapted by Davis (1975) for activity-on-the-node representation. A number of studies in the literature (for example Kolisch et al. (1995), Alvarez-Valdès and Tamarit (1989)) seem to confirm that RCPSP problems (problem m,1|cpm|C_{max}) become easier with increasing values of the *CNC*. Elmaghraby and Herroelen (1980) already questioned the use of the *CNC*. De Reyck and Herroelen (1996) reached the conclusion that it is very ambiguous to attach all explanatory power of problem complexity to *CNC*; a positive correlation exists between the *CNC* and the so-called *complexity index* (*CI*) or the *reduction complexity*, defined by (Bein et al. (1992)) as the minimum number of node reductions sufficient (along with serial and parallel reductions) to reduce a two-terminal acyclic network to a single edge.

Another well-known measure of the topological structure of an activity network is the *order strength* (*OS*), defined by Mastor (1970) as the number of precedence relations, including the transitive ones, divided by the theoretical maximum of such precedence relations, namely $n(n-1)/2$, where n denotes the number of activities. It is known that a continuous hard-easy complexity pattern exists for the resource-constrained project scheduling problem (problem m,1|cpm|C_{max}), the resource-constrained project scheduling problem with generalized precedence relations (problem m,1|gpr|C_{max}), and the discrete time/resource trade-off problem (problem 1,1|$cpm,disc,mu$|C_{max}). A continuous easy-hard complexity pattern has been observed for the unconstrained *npv* problem with finish-start precedence relations (problem cpm,c_j|npv) and generalized precedence relations with minimal and maximal timelags (problem gpr,c_j|npv).

The *complexity index* (*CI*) reveals a continuous hard-easy complexity pattern for the resource-constrained project scheduling problem with finish-start (problem m,1|cpm|C_{max}) and generalized precedence relations (problem m,1|gpr|C_{max}). A continuous easy-hard complexity pattern has been observed for the problem of generating the complete time/cost trade-off curve (problem $1,T$|$cpm,disc,mu$|$curve$) and the discrete time/resource trade-off problem (problem 1,1|$cpm,disc,mu$|C_{max}).

The popular PSPLIB instances (Kolisch and Sprecher (1996)) that are currently used by the project scheduling research community as an almost standard test set for the RCPSP, have been generated using only the *CNC* as network structure metric. The networks in the set cannot be called strongly random because they do not guarantee that the topology is a random selection from the space of all possible networks which satisfy the specified input parameters. In order to protect the test ensemble from a possible bias in network structure, researchers should be encouraged to generate test instances using problem generators that rely on other more reliable network structure metrics such as *OS* (*ProGen/max* and *RanGen*) and *CI* (*RanGen*), or even better, that generate networks with a strongly random structure (*RanGen*).

4.3.2 Resource-based parameters

ProGen (Kolisch et al. (1995)) and *ProGen/max* (Schwindt (1996)) use the resource factor *RF*. This parameter, introduced by Pascoe (1966) can be calculated as $RF = \frac{1}{nK}\sum_{i=1}^{n}\sum_{k=1}^{K}\begin{cases} 1, \text{ if } r_{ik} > 0 \\ 0, \text{ otherwise} \end{cases}$, where n denotes the number of activities (excluding dummy activities), K denotes the number of resource types, and r_{ik} denotes the amount of resource type k required by activity i. The resource factor reflects the average portion of resource types requested per activity and consequently measures the density of the matrix $\mathbf{r}_{ik}$. According to Kolisch et al. (1995), there is a positive relation between the CPU time needed to solve the RCPSP and the *RF* while Alvares-Valdès and Tamarit (1989) have observed that problems with $RF = 1.0$ were easier to solve than problems with $RF = 0.5$. It should be observed that when implementing *RF* as defined above, it is possible that no resource requirement will be generated for some activities. This is certainly true when $RF < 1/$(number of resources), but it can also happen in other cases (e.g. $RF = 0.5$ and half the number of activities use all resource types while the other half do not require any resources). *ProGen/max* uses a lower bound equal to $1/$(number of resources) for *RF* to assure that all activities use at least one resource type. *RanGen* allows for the use of *RF* as defined above, but also relies on a new measure of resource density *RU* that varies between zero and the number of resource types available and measures for each activity the number of resource types used as $RU_i = \begin{cases} 1, \text{ if } r_{ik} > 0 \\ 0, \text{ otherwise} \end{cases}$, $i = 1, ..., n$. In *RanGen* $RU_i = RU$, where *RU* is a positive constant, to assure that each activity uses at least one resource type. In doing so, the impact of the number of resources on problem hardness can be studied by varying the number of resource types K for the set of networks with $RF = 1$ (or *RU* equal to the number of resource types).

Kolisch et al. (1995) define the *resource strength* (RS_k) as $(a_k - r_k^{\min})/(r_k^{\max} - r_k^{\min})$, where a_k is the total availability of renewable resource type k, $r_k^{\min} = \max_{i=1,\ldots,n} r_{ik}$ (the maximum resource requirement for each resource type), and $r_k^{\max}$ is the peak demand for resource type k in the precedence-based early start schedule. Elmaghraby and Herroelen (1980) were the first to conjecture that the relationship between the complexity of a RCPSP and the resource availability varies according to a bell-shaped curve. De Reyck and Herroelen (1996) confirmed this conjecture and rejected the negative correlation between problem difficulty and the *RS* as found by Kolisch et al. (1995) for the PSPLIB

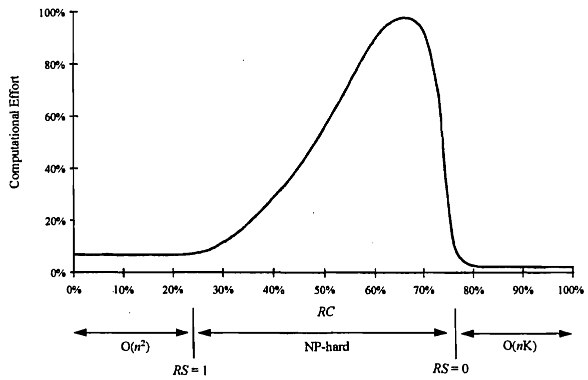

Figure 3: Computational complexity versus RS and RC

instances that have been generated using RS as a resource-based parameter.

In contrast to RS, the *resource-constrainedness* (RC_k), defined by Patterson (1976) for each resource k as $\overline{r}_k/a_k$, where a_k is the availability of resource type k and $\overline{r}_k$ is the average quantity of resource k demanded when required by an activity, is a 'pure' measure of resource availability in that it does not incorporate information about the precedence structure of a network. Moreover, there are occasions where RS can no longer distinguish between easy and hard instances while RC continues to do so.

It should be noted that both the resource strength RS_k and resource-constrainedness RC_k are defined for each renewable resource type. Hence, their unambiguous use is restricted to the case $k = 1$ or the case where RS_k and RC_k are constant over all k. When this is not the case and the RS- and RC-values would be averaged over all resource types, serious bias may be introduced in the results.

De Reyck and Herroelen (1996) observed a relatively sharp easy-hard-easy phase transition for the resource-constrained project scheduling problem (see Figure 3). They observed that the average complexity is high at the phase transition boundary, but so is the variance. Hence, they were forced to conclude that the currently used resource availability measures are too crude. An unambiguous resource availability measure does not yet exist and remains a valid topic for further research.

5 Managing project uncertainty

Most resource-constrained project scheduling research efforts have been made under the assumption that the project scheduling world is deterministic. Over the recent years, a growing literature on resource-constrained project scheduling under uncertainty has emerged (for a review we refer to Herroelen and Leus (2004ab)).

De Meyer et al. (2002) observed that managers consistently fail to recognize that there are different types of uncertainty, each of which requires a different management approach. They identified the need to set up a *project uncertainty profile*, i.e. a qualitative characterization of the degree to which each type of uncertainty may affect the project. Pich et al. (2002) show that classic project management methods emphasize *adequate information* and *instructionism*. Classical project planning approaches assume adequate project information, i.e. they assume that all possible events may be anticipated although their occurrence may be stochastic. They concentrate on two types of uncertainty: variation (many small influences with unchanging structure of the project network) and foreseen uncertainties (identifiable and understood influences that the project team cannot be sure will occur and the incorporation of chance nodes in the network allowing to deal with stochastic network evolution). All of them are a variant of *instructionism* - the ex ante determination of actions or policies (*known unknowns*) in which preplanned actions are triggered by signals.

5.1 Dealing with known unknowns

Two fundamental approaches have been used in the literature to deal with known unknowns: the use of scheduling policies and the use of proactive/reactive scheduling.

5.1.1 Stochastic project scheduling

Stochastic project scheduling deals with the problem of how to schedule project activities with uncertain durations in order to minimize the expected project duration under finish-start precedence and renewable resource constraints. No baseline schedule is constructed but so-called *scheduling policies* are used to decide at each stage of a multi-stage decision process which activities selected from the set of precedence and resource feasible activities (admissibility constraint) have to be started. The non-anticipativity constraint requires that these decisions can only be based on the observed past and a priori knowledge about processing time distributions (Möhring et al. (1984, 1985), Stork (2001)).

5.1.2 Proactive/reactive scheduling

At this juncture, the fundamental question to be asked is whether one needs a baseline schedule or not. A *baseline schedule* allows for (a) the allocation of resources to different activities, (b) quoting competitive and reliable due dates, (c) scheduling the activities in accord with all parties within the inbound and

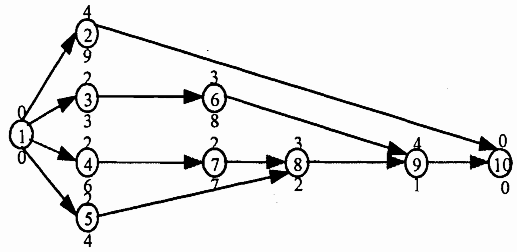

Figure 4: Project network example (Wiest and Levy 1977)

outbound supply chain, (d) agree on time windows for work to be done by subcontractors, (e) share production schedules with suppliers on a continuous basis using Internet technology, (f) making cash flow projections, (g) measure the performance of both management and shop floor personnel, and (h) project monitoring and control.

An online survey among the PMI member community (*PM Network*, July 2003, p. 12) revealed that only 5% of the 59 respondents cited *unexpected risk* as the factor that most severely impacts the ability to deliver projects on time and on budget. The other cited factors (unrealistic estimates/milestones for 42%, lack of executive support (29%) and scope changes (25%)) are recognizable sources of uncertainty, so that risks arising from them are not unexpected, but can be identified, assessed and managed proactively (Hillson (2003)).

Chapman and Ward (2003) state that proactive and reactive planning are not alternatives, but are complementary aspects of planning as a whole. Proactive/reactive scheduling approaches generate a baseline schedule that incorporates a degree of anticipation of variability during project execution and/or information about the reactive scheduling approach to be used. The idea is to generate a schedule that is (a) sufficiently *stable* (*solution robust*) in that the planned activity start times are rather insensitive to changes in the input data, and (b) sufficiently *quality robust* in that the schedule performance (objective value) is rather insensitive to disturbances that may occur during schedule execution.

Critical chain Over the past few years, the *critical chain* (*CC*) methodology (Goldratt (1997)) has received a lot of attention. As already mentioned above, the methodology has been turned into commercial software, either as an add-in to the popular project planning software package Microsoft Project® or as an integrated module of the project planning package itself.

The basic scheduling methodology of *CC* can best be illustrated on the project example shown in Figure 4. The project is shown in activity-on-the-node

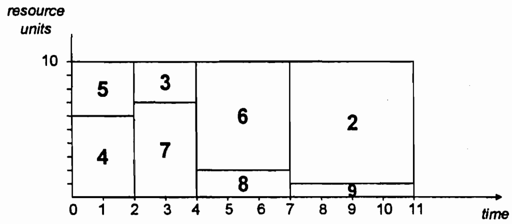

Figure 5: Minimum makespan schedule

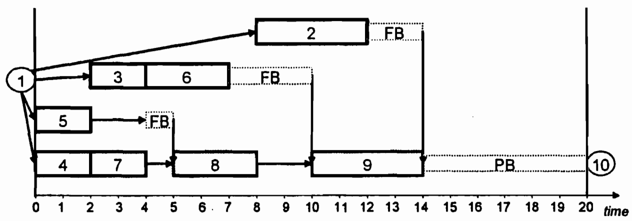

Figure 6: Buffered schedule

representation. Node 1 and node 10 refer to a dummy start activity and a dummy end activity, respectively. The number shown above a node represents the planned duration of the corresponding activity, the number shown below a node denotes the per period requirement (number of units) for a single renewable resource that is available in a constant amount of 10 units. Figure 5 shows a minimum makespan solution for the corresponding RCPSP. The schedule shows 16 different critical chains (a *critical chain* is defined as the longest chain of precedence dependent and/or resource dependent activities that determines the makespan). ProChain®, one of the popular Microsoft Project® add-ins, computes the chain <1-4-7-8-9-10> as the critical chain with a length of 11 time periods.

Figure 6 shows the buffered baseline schedule generated by ProChain®. The software has computed the critical chain as the chain <1-4-7-8-9-10> and has applied the 50% buffer sizing rule to add a 6-period *project buffer.* This project buffer should protect the project due date (here time instant 20) against variation in the critical chain activities. In this sense, *CC* aims at generating a *makespan protecting buffered schedule.* The software has also introduced *feeding buffers* where non-critical chain activities merge with the critical chain: a one-period feeding buffer FB 5-8 to protect the critical chain against disruptions in activity 5, a three-period feeding buffer FB 6-9 to protect the critical chain against variation in the feeding chain <3-6>, and a two-period feeding buffer following activity 2 to protect the critical chain against variation in activity 2.

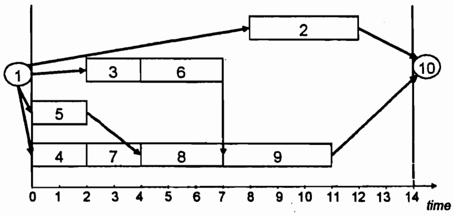

Figure 7: Projected schedule

In this case, this feeding buffer is inserted in front of the project buffer. The reader will also observe that the critical chain is no longer a chain (it has two gaps) while there is actually no reason to have a gap between activity 7 and activity 8 and between activity 8 and 9. The idea is that the buffers should act as a proactive protection mechanism during project execution, generating warning signals when buffers are penetrated beyond a prespecified percentage.

The software also generates a so-called *projected schedule,* shown in Figure 7, to be used during project execution. The projected schedule is not buffered and applies the so-called roadrunner mentality: except for the so-called gating tasks (tasks with no real predecessors), activities are scheduled as early as possible. As can be seen, activities 8 and 9 are left-shifted allowing the gaps in the critical chain to be closed, while the starting time of the gating task 2 is left unchanged. The projected schedule is to be recomputed when distortions occur. Similar to the unprotected minimum duration schedule of Figure 5, it is not stable (solution robust) in that the slightest distortion in many activity start times will have an immediate impact on the planned starting times of others.

Herroelen and Leus (2001) have validated the working principles of *CC*. They reached the conclusion that *CC* acted as an important eye-opener but constitutes a serious oversimplification of the real problem and induces the need for additional research.

Figures 8 and 9 reveal some important idiosyncrasies of *CC*. It is evident that the critical chain is schedule dependent. Figure 8 shows the 20-period schedule obtained using the early start priority rule. If the chain <1-2-3-5-6-7-8-9-10> is chosen as the critical chain, the corresponding buffered schedule of Figure 9 is obtained. The planned duration is inflated to a total of 30 periods and the feeding buffer FB 4-7 clearly fails to act as a proactive mechanism. The slightest prolongation of activity 4 will not immediately generate a buffer penetration alert, though it immediately leads to a resource conflict with activity 6 and as a result immediately disrupts the critical chain.

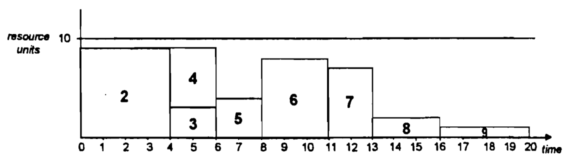

Figure 8: Feasible schedule obtained using the early start time priotity rule

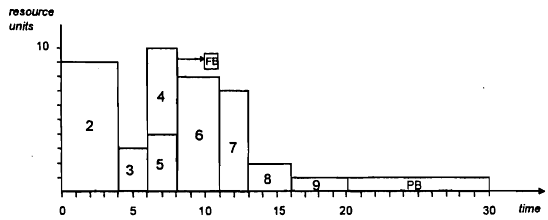

Figure 9: Buffered schedule

Proactive/reactive scheduling Research on proactive/reactive project scheduling methods is just in its burn-in phase. Exact and suboptimal procedures for generating stable schedules have been developed by Leus (2003), Herroelen and Leus (2004ac) and Leus and Herroelen (2004). Assuming for our problem example a project deadline of 14, equal activity disruption probabilities, a 50% chance for a one period activity duration increase and a 50% chance for a two-period activity duration increase, their suggested methodology allows for the generation of the stable schedule shown in Figure 10, under the assumption that the expected deviation of the realized activity starting times from the planned activity starting times is used as a stability measure. In comparison to the projected schedule of Figure 7, this schedule has identical makespan but has gained in solution robustness.

Van de Vonder et al. (2004) have addressed the potential trade-off between the quality robustness (measured in terms of project duration) and solution

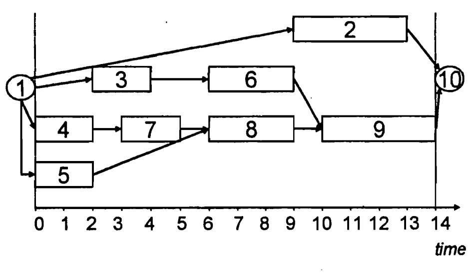

Figure 10: Stable schedule

robustness (measured in terms of the deviation between the planned and realized start times). The authors reach the conclusion that the expected difference in makespan performance between makespan protecting schedules (such as CC) and solution robust schedules tends to disappear for some projects. Where this is the case, a solution robust schedule will most likely be preferred because of the considerably lower stability costs.

Exploring flexibility Flexibility refers to the freedom to change schedules during the project execution phase. The changes may have to do with the timing, sequence, resource allocation, activity execution modes, etc. An interesting research track is the exploration of so-called *group sequences* in the context of contingency scheduling. Billaut and Roubellat (1996) suggest to generate for every resource a group sequence, i.e. a totally or partially ordered set of groups of operations, and to consider all the schedules obtained by an arbitrary choice of the ordering of the operations inside each group. In this way, the decision maker is not provided with one feasible schedule but with several ones. The hope is that during the real-time execution of the schedule, it becomes possible to switch from one solution to the other in the presence of a disruption without any loss in performance (see e.g. Artigues et al. (1999)).

5.2 Dealing with unknown unknowns

Pich et al. (2002) remark that an important challenge for a project team that is not sufficiently addressed in the existing project management literature is dealing with inadequate project information. Unforeseen uncertainties (*unknown unknowns* or *unk-unks*) cannot be identified during project planning. Contingencies cannot be planned for unk-unks. *Information inadequacy* can arise from both project ambiguity and project complexity. *Ambiguity* refers to the lack of awareness of the project team about certain states of the world or causal relationships. *Project complexity* means that different actions and states of the world parameters interact so that the effect of actions is difficult to assess. The authors discuss two fundamental strategies that can be used: *learning* (scanning to identify unk-unks and problem solving to modify policies) and *selectionism* (pursuing multiple approaches and choosing the best one ex post). Clearly, more research is needed to refine the suggested management approaches and to perform empirical tests.

6 Summary and conclusions

The results of several field studies and surveys conducted among project management practitioners reveal that there is still a wide gap between project management practice and the state of development of project scheduling theory. Projects are frequently prone to escalation in time and budget. The lack of adequate planning and control is often cited as one of the major variables that best distinguish between escalated and nonescalated projects. Many project scheduling procedures published over the past several years in the open literature have not yet found their way to the commercial software and are seldom or not used by practicing project schedulers. Studies reveal that managers often use information systems for project planning mainly for communication and representation, rather than for optimization. Moreover, software users seem to have limited knowledge of the software tool they are using and project planning tools in general. Commercial software packages rely on simple priority rule based heuristics for dealing with deterministic resource levelling and resource-constrained project scheduling problems. Most research efforts have concentrated on single project scheduling, while surveys indicate that multi-project settings dominate in practice.

The hierarchical framework for project planning and control presented in this text allows at each of its hierarchical levels for the application of a positioning scheme that identifies suitable single and multi-project scheduling methods taking into account both variability and dependency of the project environment.

As for the deterministic resource-constrained project scheduling problems, our analysis revealed the need for additional research on branch-and-bound and branch-and-cut solution procedures. The development of hybrid metaheuristics that incorporate more artificial intelligence and operations research techniques is a viable area of future research.

The current practice of validating exact and heuristic scheduling procedures on test instances generated by commonly used problem generators reveals the need for additional research. The instances generated by some generators do not possess a strongly random network structure while an unambiguous resource availability measure does not yet exist.

Most resource-constrained project scheduling research efforts have been made under the assumption that the project scheduling world is deterministic, while uncertainties during project execution are quite common. Two fundamental approaches have been used in the literature to deal with known unknowns: the use of scheduling policies and the use of proactive/reactive project scheduling. The former methods do not rely on a baseline schedule but decide at each stage of a multi-stage decision process which activities are to be started. Over the recent years we have witnessed the introduction of a number of software implementations of the critical chain methodology. We have revisited some of the merits and pitfalls of this approach and have identified some interesting topics for future research.

It is our hope that the many research tracks identified in this paper may help to close the gap between project scheduling theory and practice.

Acknowledgement

This research has been partially supported by project OT/03/14 of the Research Fund K.U.Leuven and project G.0109.04 of the Research Programme of the Fund for Scientific Research - Flanders (Belgium) (F.W.O.-Vlaanderen).

References

Alvarez-Valdès, R. and Tamarit, J.M. (1989). Heuristic algorithms for resource-constrained project scheduling. In Slowinski, R. and Weglarz, J. (Eds.). *Advances in project scheduling*, 134-143. Elsevier, Amsterdam.

Artigues, C., Roubellat, F. and Billaut, J.-C. (1999). Characterization of a set of schedules in resource-constrained multi-project scheduling problem with multiple modes. *International Journal of Industrial Engineering - Theory, applications and practice*, **6**(2), 112-122.

Baptiste, P. and Demassey, S. (2004). Tight LP bounds for resource constrained project scheduling. *OR Spectrum*, **26**(2), 251-262.

Baptiste, P. and Le Pape, C. (2000). Constraint propagation and decomposition techniques for highly disjunctive and highly cumulative project scheduling problems. *Constraints*, **5**, 119-139.

Baptiste, P., Le Pape, C. and Nuijten, W. (1999). Satisfiability tests and time-bound adjustments for cumulative scheduling problems. *Annals of Operations Research*, **92**, 305-333.

Bein, W.W., Kamburowski, J. and Stallmann, M.F.M. (1992). Optimal reduction of two-terminal directed acyclic graphs. *SIAM Journal on Computing*, **21**, 1112-1129.

Belassi, W. and Icmeli Tukel, O. (1996). A new framework for determining critical success/failure factors in projects. *International Journal of Project Management*, **14**(3), 141-151.

Bigelow, D. (2004). The future is EPM. *PM Network*, April, 24-26.

Billaut, J.C. and Roubellat, F. (1996). Characterization of a set of schedules in a multiple resource context. *Journal of Decision Systems*, **5**(1-2), 95-109.

Bock, D.B. and Patterson, J.H. (1990). A comparison of due date setting, resource assignment, and job preemption heuristics for the multi-project scheduling problem. *Decision Sciences*, **21**, 387-402.

Bounds, G. (1998). The last word on project management. *IIE Solutions*, **30**(11), 41-43.

Brucker, P., Drexl, A., Möhring, R., Neumann, K. and Pesch, E. (1999). Resource-constrained project scheduling: notation, classification, models and methods. *European Journal of Operational Research*, **112**, 3-41.

Brucker, P. and Knust, S. (2000). A linear programming and constraint propagation-based lower bound for the RCPSP. *European Journal of Operational Research*, **127**, 355-362.

Brucker, P., Knust, S., Schoo, A. and Thiele, O. (1998). A branch & bound algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, **107**, 272-288.

Bryde, D.J. (2003). Project management concepts, methods and application. *International Journal of Project Management*, **23**(7), 775-793.

Carlier, J. and Néron, E. (2000). A new LP-based lower bound for the cumulative scheduling problem. *European Journal of Operational Research*, **127**, 355-362.

Chapman, C. and Ward, S. (2003). *Project risk management - Processes, techniques and insights*, 2nd edition. John Wiley & Sons.

Cheeseman, P., Kanefsky, B. and Taylor W.M. (1991). Where the really hard problems are. *Proceedings of the International Joint Conference of Artificial Intelligence*, **1**, 331-337.

Christofides, N., Alvarez-Valdès, R. and Tamarit, J.M. (1987). Project scheduling with resource constraints: a branch and bound approach. *European Journal of Operational Research*, **29**, 262-273.

Coelho, J.S. (2004). Generating RCPSP instances with known optimal solutions. *Proceedings of PMS 2004* (Oulamara, A. and Portmann, M.-C. (Eds.)), 70-75.

C/S Solutions, Inc. (2002). Risk+ user's guide.

Damay, J., Quilliot, A. and Sanlaville, E. (2004). RCPSP: new approach, new gaps. *Proceedings of PMS 2004* (Oulamara, A. and Portmann, M.-C. (Eds.)), 62-65.

Danna, E., Rothberg, E. and Le Pape, C. (2004). Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming*, to appear.

Davis, E.W. (1975). Project network summary measures and constrained resource scheduling. *IIE Transactions*, **7**, 132-142.

De Boer, R. (1998). Resource-constrained multi-project management - A hierarchical decision support system. PhD thesis, University of Twente, Enschede.

De Meyer, A., Loch, C.H. and Tisch, M. (2002). Managing project uncertainty: from variation to chaos. *MIT Sloan Management Review*, Winter, 60-67.

De Reyck, B. and Herroelen, W. (1996). On the use of the complexity index as a measure of complexity in activity networks. *European Journal of Operational Research*, **91**, 347-366.

De Reyck, B. and van de Velde, S. (1999). Informatiesystemen voor projectplanning: meer communicatie dan optimalisatie. *Business Logistics*, **99**(10), 104-110.

De Wit, J. and Herroelen, W. (1990). An evaluation of microcomputer-based software packages for project management. *European Journal of Operational Research*, **49**, 102-139.

Debels, D., De Reyck, B., Leus, R. and Vanhoucke, M. (2004). A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. *Proceedings of PMS 2004* (Oulamara, A. and Portmann, M.-C. (Eds.)), 23-26.

Deckers, M. (2001). Exploratief onderzoek naar het gebruik van informatiesystemen voor projectplanning. Eindverhandeling. Department of Applied Economics, K.U.Leuven.

Demassey, S., Artigues, C., Baptiste, Ph. and Michelon, P. (2004). Lagrangean relaxation-based lower bounds for the RCPSP. *Proceedings of PMS 2004* (Oulamara, A. and Portmann, M.-C. (Eds.)), 76-79.

Demassey, S., Artigues, C. and Michelon, P. (2003). Constraint propagation based cutting planes: an application to the resource-constrained project scheduling problem. Technical Report 237, Laboratoire d'Informatique d'Avignon, France.

Demeulemeester, E. and Herroelen, W. (1992). A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management Science*, **38**, 1803-1818.

Demeulemeester, E. and Herroelen, W. (1997). New benchmark results for the resource-constrained project scheduling problem. *Management Science*, **43**, 1485-1492.

Demeulemeester, E. and Herroelen, W. (2002). *Project scheduling - A research handbook.* Vol. 49 of International Series in Operations Research & Management Science. Kluwer Academic Publishers, Boston.

Demeulemeester, E.L., Vanhoucke, M. and Herroelen, W.S. (2003). RanGen: A network generator for activity-on-the-node networks. *Journal of Scheduling*, **6**, 17-38.

Dorndorf, U., Pesch, E. and Phan-Huy, T. (2000). A time-oriented branch-and-bound algorithm for project scheduling with generalized precedence constraints. *Management Science*, **46**, 1365-1384.

Dumond, J. (1992). In a multi-resource environment, how much is enough. *International Journal of Production Research*, **30**(2), 395-410.

Dumond, J. and Mabert, V.A. (1988). Evaluating project scheduling and due date assignment procedures: an experimental analysis. *Management Science*, **34**, 101-118.

Elmaghraby, S.E.E. (2002). Contribution to the round table discussion on new challenges in project scheduling (PMS2002), Valencia, Spain, April 3-5.

Elmaghraby, S.E.E. and Herroelen, W.S. (1980). On the measurement of complexity in activity networks. *European Journal of Operational Research*, **5**, 223-234.

Feldman, E.J. (1985). *Concorde and dissent - Explaining high technology failures in Britain and France.* Cambridge University Press.

Fischetti, M. and Lodi, A. (2003). Local branching. *Mathematical programming*, **98**(1-3), 23-47.

Fleszar, K. and Hindi, K.S. (2004). Solving the resource-constrained project scheduling problem by a variable neighbourhood search. *European Journal of Operational Research*, **155**, 402-413.

Flowers, S. (1996). *Software failure: management failure.* John Wiley and Sons, Chichester.

Gademann, A.J.R.M. and Schutten, J.M.J. (2001). Linear programming based multi-project heuristics for multi-project capacity planning. Working paper TBK01W-004 OMST-002, University of Twente, Enschede.

Gantthead (2003). Why projects succeed and fail. //www.gantthead.com.

Glass, R.L. (1998). *Software runaways.* Prentice-Hall, Inc., N.J.

Glass, R.L. (1999). *Computing calamities.* Prentice-Hall, Inc., N.J.

Goldratt, E. M. (1997). *Critical chain.* The North River Press, Great Barrington, U.S.A.

Hall, P. (1982). *Great planning disasters.* University of California Press.

Hans, E.W. (2001). Resource loading by branch-and-price techniques. PhD thesis, University of Twente, Enschede.

Hans, E.W., Herroelen, W., Leus, R. and Wullink, G. (2003). A hierarchical approach to multi-project planning under uncertainty. Research Report 0346, Department of Applied Economics, K.U.Leuven, Belgium.

Hartmann, S. and Kolisch, R. (2000). Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*, **127**, 394-407.

Hayes, B. (1997). Can't get no satisfaction. *American Scientist*, **85**, 108-112.

Elmaghraby, S.E.E. and Herroelen, W.S. (1980). On the measurement of complexity in activity networks. *European Journal of Operational Research*, **5**, 223-234.

Herroelen, W., De Reyck, B. and Demeulemeester, E. (1998a). Resource-constrained project scheduling - A survey of recent developments. *Computers and Operations Research*, **25**(4), 279-302.

Herroelen, W. and De Reyck, B. (1999). Phase transitions in project scheduling. *Journal of the Operational Research Society*, **50**, 148-156.

Herroelen, W., Demeulemeester, E. and De Reyck, B. (1998b), A classification scheme for project scheduling. Chapter 1 in Weglarz, J. (Ed.). *Project scheduling - Recent models, algorithms and applications.* International Series in Operations Research and Management Science, **14**, 77-106, Kluwer Academic Publishers.

Herroelen, W. and Leus, R. (2001). On the merits and pitfalls of critical chain scheduling. *Journal of Operations Management*, **19**, 557-577.

Herroelen, W. and Leus, R. (2004a). Project scheduling under uncertainty, survey and research potentials. *European Journal of Operational Research*, to appear.

Herroelen, W. and Leus, R. (2004b). Robust and reactive project scheduling: a review and classification of procedures. *International Journal of Production Research*, **42**(8), 1599-1620.

Herroelen, W. and Leus, R. (2004c). The construction of stable project baseline schedules. *European Journal of Operational Research*, **156**(3), 550-565.

Herroelen, W.S., Van Dommelen, P. and Demeulemeester, E.L. (1997). Project management with discounted cash flows - A guided tour through recent developments. *European Journal of Operational Research*, **100**(1), 97-121.

Hillson, D. (2003), Addressing risk. *PM Network*, October, 6.

Hubermann, B.A. and Hogg, T. (1987). Phase transitions in artificial intelligence systems. *Artificial Intelligence*, **33**, 155-171.

Icmeli-Tukel, O. and Rom, W.O. (1997). Ensuring quality in resource-constrained project scheduling. *European Journal of Operational Research*, **103**, 483-496.

ILOG (2002). ILOG CPLEX 7.5, Reference Manual. ILOG, Inc., Mountain view, California.

Johnson, R.V. (1992). Resource-constrained scheduling capabilities of commercial project management software. *Project Management Journal*, **22**, 39-43.

Johnson, J., Boucher, K.D., Connors, K. and Robinson, J. (2001). Project management: The criteria for success. *Software Magazine*, **21**(1), s3-s11.

Kavadias, S. and Loch, C.H. (2004). *Project selection under uncertainty - Dynamically allocating resources to maximize value.* Kluwer Academic Publishers, Boston.

Keil, M., Rai, A., Mann, J.E.C. and Zhang, G.P. (2003). Why software projects escalate: The importance of project management constructs. *IEEE Transactions on Engineering Management*, **50**(3), 251-261.

Kharbanda, O.M. (1983). *How to learn from project disasters.* Ashgate Publishing Company.

Klein, R. (2000). *Scheduling of resource-constrained projects.* Kluwer Academic Publishers, Boston.

Klein, R. and Scholl, A. (1999). Computing lower bounds by destructive improvement: an application to resource-constrained project scheduling. *European Journal of Operational Research*, **112**(2), 322-346.

Kolisch, R. (1999). Resource allocation capabilities of commercial project management software packages. *Interfaces*, **29**(4), 19-31.

Kolisch, R. and Hartmann, S. (1999). Heuristic algorithms for solving the resource-constrained project scheduling problem: classification and computational analysis. In Weglarz, J. (Ed.). *Project scheduling - Recent models, algorithms and applications*, 147-178, Kluwer Academic Publishers, Boston.

Kolisch, R. and Padman, R. (2001). An integrated survey of deterministic project scheduling. *Omega*, **29**, 249-272.

Kolisch, R. and Sprecher, A. (1996). PSPLIB - A project scheduling library. *European Journal of Operational Research*, **96**, 205-216.

Kolisch, R., Sprecher, A. and Drexl, A. (1995). Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, **41**(10), 1693-1703.

Kurtulus, I. (1985). Multi-project scheduling: analysis of scheduling strategies under unequal delay penalties. *Journal of Operations Management*, **5**(3), 291-303.

Kurtulus, I.S. and Davis, E.W. (1982). Multi-project scheduling: categorization of heuristic rules performance. *Management Science*, **28**(2), 161-172.

Kurtulus, I.S. and Narula, S.C. (1985). Multi-project scheduling: analysis of project performance. *IIE Transactions*, **17**(1), 58-66.

Lawrence, S.R. and Morton, T.E. (1993). Resource-constrained multi-project scheduling with tardy costs: comparing myopic bottleneck and resource pricing heuristics. *European Journal of Operational Research*, **64**, 168-187.

Leus, R. (2003). The generation of stable project plans. Complexity and exact algorithms. Ph.D. thesis, Katholieke Universiteit Leuven, Belgium.

Leus, R. and Herroelen, W. (2004). Stability and resource allocation in project planning. *IIE Transactions*, **156**(3), 550-565.

Liberatore, M.J. and Pollack-Johnson, B. (2003). Factors influencing the usage and selection of project management software. *IEEE Transactions on Engineering Management*, **50**(2), 164-174.

Liberatore, M.J., Pollack-Johnson, B. and Smith, C.A. (2001), Project management in construction: software use and research directions. *Journal of*

Construction Engineering and Management, March/April, 101-107.

Lova, A., Maroto, C. and Tormos, P. (2000). A multicriteria heuristic method to improve resource allocation in multi-project scheduling. *European Journal of Operational Research*, **127**, 408-424.

Lova, A. and Tormos, P. (2002). Combining random sampling and backward-forward heuristics for resource-constrained multi-project scheduling. Proceedings of the Eight International Workshop on Project Management and Scheduling, Valencia, April 3-5, 244-248.

Maroto, C. and Tormos, P. (1994). Project management: an evaluation of software quality. *International Transactions in Operational Research*, **1**(2), 209-221.

Maroto, C., Tormos, P. and Lova, A. (1999). The evolution of software quality in project scheduling. In *Project scheduling - Recent models, algorithms and applications* (Weglarz, J. (ed.)), Chapter 11, Kluwer Academic Publishers, Boston.

Mastor, A.A. (1970). An experimental and comparative evaluation of production line balancing techniques. *Management Science*, **16**, 728-746.

Milis, K. and Mercken, R. (2002). Success factors regarding the implementation of ICT investment projects. *International Journal of Production Economics*, **80**, 105-117.

Mingozzi, A., Maniezzo, V., Ricciardelli, S. and Bianco, L. (1998). An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation. *Management Science*, **44**, 715-729.

Möhring, R.H., Radermacher, F.J. and Weiss, G. (1984). Stochastic scheduling problems I - general strategies. *ZOR - Zeitschrift für Operations Research*, **28**, 193-260.

Möhring, R.H., Radermacher, F.J. and Weiss, G. (1985). Stochastic scheduling problems II - set strategies. *ZOR - Zeitschrift für Operations Research*, **29**, 65-104.

Möhring, R.H., Schulz, A.S., Stork, F. and Uetz, M. (2003). Solving project scheduling problems by minimum cut computations. *Management Science*, **49**(3), 330-350.

Morris, P.W.G. and Hough, C.H. (1988). *The anatomy of major projects - A study of the reality of project management.* John Wiley & Sons.

Morton, T.E. and Pentico, D.W. (1993). *Heuristic scheduling systems - With applications to production systems and project management.* John Wiley & Sons, Inc., New York.

Neumann, K., Schwindt, C. and Zimmermann, J. (2003). *Project scheduling with time windows and scarce resources.* Springer-Verlag. Heidelberg.

Özdamar, L. and Ulusoy, G. (1996). A survey of the resource-constrained project scheduling problem. *IIE Transactions*, **27**(5), 574-586.

Pascoe, T.L. (1966). Allocation of resources - CPM. *Revue Française de Recherche Opérationnelle*, **38**, 31-38.

Patterson, J.H. (1976). Project scheduling: The effects of problem structure on heuristic performance. *Naval Research Logistics*, **23**, 95-123.

Payne, J.H. (1995). Management of multiple simultaneous projects: a state-of-the-art review. *International Journal of Project Management*, **13**(3), 163-168.

Pich, M., Loch, C.H. and De Meyer, A. (2002). On uncertainty, ambiguity, and complexity in project management. *Management Science*, **48**(8), 1008-1023.

Pinto, J.K. (2002). Project management 2002. *Research Technology Management*, March-April , 22-37.

Pinto, J.K. and Mantel, S. (1990). The causes of project failure. *IEEE Transactions on Engineering Management*, **EM-37**, 269-276.

Pinto, J.K. and Prescott, J.E. (1990). Planning and tactical factors in the project implementation process. *Journal of Management Studies*, **27**(3), 305-327.

Pinto, J.K. and Slevin, D.P. (1988). Critical success factors across the project lifecycle. *Project Management Journal*, **19**(3), 67-75.

Pollack-Johnson, B. and Liberatore, M.J. (1998). Project management software usage patterns and suggested research directions for future developments. *Project Management Journal*, **20**(2), 19-28.

Sankaran, J.K., Bricker, D.L. and Huang, S.-H. (1999). A strong fractional cutting-planes algorithm for resource-constrained project scheduling. *International Journal of Industrial Engineering*, **6**(2), 99-111.

Schonberger, R.J. (1981). Why projects are "always" late: a rationale based on manual simulation of a PERT/CPM network. *Interfaces*, **11**(5), 66-70.

Schwindt, C. (1996). Generation of resource-constrained project scheduling problems with minimal and maximal timelags. Report WIOR-489, Institut für Wirtschaftstheorie und Operations Research, Universität Karlsruhe, Germany.

Söderlund, J. (2004). Building theories of project management: past research, questions for the future. *International Journal of Project Management*, **22**, 183-191.

Sprecher, A. (2000). Scheduling resource-constrained projects competitively at modest resource requirements. *Management Science*, **46**, 710-723.

Standish Group, The (1994). The CHAOS Report. Report available at http://www.standishgroup.com/sample_research/chaos_1994_1.php.

Stork, F. (2001). Stochastic resource-constrained project scheduling. Ph.D. thesis, Technische Universität Berlin, Germany.

Tereso, A.P., Araùjo, M.M. and Elmaghraby, S.E.E. (2004). Adaptive resource allocation in multimodal activity networks. *International Journal of Production Economics*, to appear.

Turner, J.R. (1993). *The handbook of project-based management.* Mc Graw-Hill, UK.

Valls, V., Ballestin, F. and Quintanilla, S. (2003). A hybrid genetic algorithm for the RCPSP. Research report, University of Valencia, Spain.

Van de Vonder, S., Demeulemeester, E., Herroelen, W. and Leus, R. (2004). The use of buffers in project management: the trade-off between stability and makespan. *Proceedings of PMS 2004* (Oulamara, A. and Portmann, M.-C. (Eds.)), 31-34.

Weglarz, J. (Editor) (1999). *Project scheduling - Recent models, algorithms and applications.* Kluwer Academic Publishers, Boston.

Wiest, J.D. and Levy, F.J. (1977). *A management guide to PERT/CPM with GERT/PDM/DCPM and other networks.* Prentice-Hall, Inc. Englewood Cliffs, New Jersey.

Winch, G. (1996). Thirty years of project management - What have we learned? Text available at http://bprc.warwick.ac.uk/repwinch.html.

Wullink, G., Hans, E.W., Gademann, A.J.R.M. and van Harten, A. (2003). A scenario based approach for the flexible resource loading problem. BETA WP 97. University of Twente, Enschede, The Netherlands.

Yourdon, E. (2003). *Death March*, 2nd edition. Prentice Hall, Inc: New Jersey.

Zhu, G., Bard, J.F. and Yu, G. (2003). A branch-and-cut procedure for the multi-mode resource constrained project scheduling problem. Paper presented at the INFORMS Atlanta Meeting, October.

Zwikael, O. and Globerson, S. (2004). Evaluating the quality of project planning: a model and field results. *International Journal of Production Research*, **42**(8), 1545-1556.